Proposed Standard Extension

Extended FOR .. NEXT syntax

Syntax

The syntax of the FOR .. NEXT construct should be extended to allow the control variable to assume values from a list of actual values and value ranges.

The new syntax should be (for the long form):

```
FOR <control variable> := <specifier> {,<specifier>} DO <eol>

<specifier>::= <numeric expression> [TO <numeric expression>
                                        [STEP <numeric expression>]]
```

The short form loop is constructed similarly

The syntax for the NEXT statement remains unchanged.

Semantics

The loop

```
FOR x := 1 TO 2 STEP 0.5 , 2 TO 4 , 7 , 9 , 11 DO
  <statement list>
NEXT x
```

will execute with x set to : 1,1.5,2,2,3,4,7,9,11

Note : X takes the value '2' twice, and that the value on exit from the loop is undefined (as it is at present).

Note : consider

```
FOR x := 1 DO
  <statement list>
NEXT x
```

This is a valid loop, but has the unfortunate effect that x is effectively undefined after the NEXT statement despite its apparent similarity to an assignment statement.

It is suggested that on exit from any FOR loop the value of the control variable be defined to be the TO limit of the value range provided, or its 'from' value if the loop was never executed because of incompatibility between the STEP size and the TO limit. In the case of the example above, x would be 1 on exit, and in the previous example, x would be 11 on exit.

J G Mansion

29:1:1985

```
for x=1 to 10 do
  xlar
endfor x

proc xlar
for x=1 to 20 do Rmd "x",
endproc.xlar.
```